Amendments to the Claims

This listing of claims will replace all prior versions and listings of claims in the application:

1. (original) A method for cache coherence in a distributed shared memory (DSM) multiprocessor system comprising:

(1) receiving a shared access request for information;

(2) determining whether the shared access request originates from a local node or from a remote node;

(3) processing the shared access request as an exclusive access request when the shared access request originates from the remote node; and

(4) processing the shared access request when the shared access request originates from the local node.

2. (original) The method of claim 1, wherein said step of determining whether the shared access request originates from the local node or the remote node further comprises determining a memory location where the information is stored.

3. (original) The method of claim 1, wherein said step of processing the shared access request as an exclusive access request further comprises transmitting an invalidate instruction to other local or remote nodes to eliminate copies of the requested information from other local or remote nodes.

4. (original) The method of claim 1, wherein said step of determining whether the shared access request originates from the local node or the remote node further comprises comparing

upper bits of an address of a shared access requestor with upper bits of an address of a receiving node.

5. (original) The method of claim 4, wherein said step of processing the shared access request as the exclusive access request further comprises converting the shared access request to the exclusive access request if the upper bits of the address of the shared access requestor match the upper bits of the address of the receiving node.

6. (original) The method of claim 1, wherein said step of determining whether the shared access request originates from the local node or the remote node further comprises comparing an address of a shared access requestor with a table of addresses stored by a receiving node.

7. (original) The method of claim 6, wherein said step of processing the shared access request as the exclusive access request further comprises converting the shared access request to the exclusive access request if the address of the shared access requestor matches at least one address in the table of addresses stored by the receiving node.

8. (original) The method of claim 1, wherein the method further comprises:

(5) eliminating copies of the requested information from other local or remote nodes when the shared access request is converted to the exclusive access request;

(6) storing a pointer to a shared access requestor, a location associated with the shared access request or where the requested information is cached, when the shared access request is remotely originated;

(7) determining whether the shared access request is valid;

(8) terminating an exclusive access when the exclusive access request is no longer valid; and

(9) returning information requested by the shared access request to cache.

9. (original) The method of claim 1, wherein a shared access request from a remote node has been processed as an exclusive access request, the method further comprising:

(5) receiving a subsequent shared access request from the local node at a receiving node;

(6) transmitting an invalidate instruction from the receiving node to the remote node, when the local node transmits the shared access request;

(7) terminating an exclusive access of the remote node; and

(8) processing the shared access request of the local node as the exclusive access request.

10. (original) The method of claim 9, wherein step (6) further comprises comparing upper bits of an address of a shared access requestor of the local node with upper bits of the address of the receiving node.

11. (original) The method of claim 10, wherein step (8) further comprises converting the shared access request of the local node to the exclusive access request if the upper bits of the address of the shared access requestor of the local node match the upper bits of the address of the receiving node.

12. (original) The method of claim 9, wherein step (6) of transmitting another invalidate instruction further comprises comparing upper bits of an address of the shared access request of the local node with a table of addresses stored by the receiving node.

13. (original) The method of claim 12, wherein step (8) further comprises converting the shared access request of the local node to the exclusive access request if the address of the shared access request of the local node matches at least one address in the table of addresses stored by the receiving node.

14. (original) The method of claim 9, wherein the method further comprises:

(9) storing a pointer to the shared access requestor of the local node;

(10) determining whether the shared access request of the local node is valid;

(11) terminating the exclusive access request of the local node when the exclusive access request is no longer valid; and,

(12) returning information requested by the shared access request of the local node to cache.

15. (original) A method for cache coherence in a distributed shared memory (DSM) multiprocessor system having a directory-based protocol comprising:

(1) receiving a shared access request for information by a receiving node;

(2) determining whether the shared access request originates from a local node or from a remote node;

(3) processing the shared access request as an exclusive access request when the shared access request originates from the remote node;

(4) processing the shared access request when the shared access request originates from the local node;

(5) transmitting an invalidate instruction to other local or remote nodes to eliminate copies of the requested information from the other local or remote nodes;

(6) eliminating copies of the requested information from other local or remote nodes once the shared access request is converted to the exclusive access request;

(7) storing a pointer to a shared access requestor when the shared access request is remotely originated;

(8) determining whether the shared access request is valid;

(9) terminating an exclusive status when the exclusive access request is no longer valid; and

(10) returning information requested by the shared access request to cache.

16. (original) The method of claim 15, wherein step (2) further comprises comparing upper bits of an address of the shared access requestor with upper bits of an address of the receiving node.

17. (original) The method of claim 16, wherein step (3) further comprises converting the shared access request to the exclusive access request if the upper bits of the address of the shared access requestor match the upper bits of the address of the receiving node.

18. (original) The method of claim 15, wherein step (2) further comprises comparing address of the shared access requestor with a table of addresses stored by the receiving node.

19. (original) The method of claim 18, wherein step (3) further comprises converting the shared access request to the exclusive access request if the address of the shared access requestor matches at least one address in the table of addresses stored by the receiving node.

20. (original) The method of claim 15, wherein the method further comprises:

(11) receiving a shared access request from the local node;

(12) transmitting an invalidate instruction from the receiving node to the remote node, when the local node transmits the shared access request;

(13) terminating the exclusive access of the remote node;

(14) processing the shared access request of the local node as the exclusive access request.

21. (original) The method of claim 20, wherein step (12) further comprises comparing upper bits of an address of the shared access requestor of the local node with upper bits of the address of the receiving node.

22. (original) The method of claim 21, wherein step (14) further comprises converting the shared access request of the local node to the exclusive access request if the upper bits of the address of the shared access requestor of the local node match the upper bits of the address of the receiving node.

23. (original) The method of claim 20, wherein step (12) further comprises comparing upper bits of an address of the shared access requestor of the local node with a table of addresses stored by the receiving node.

24. (original) The method of claim 23, wherein step (14) further comprises converting the shared access requestor of the local node to the exclusive access request if the address of the shared access requestor of the local node matches at least one address in the table of addresses stored by the receiving node.

25. (original) The method of claim 20, wherein the method further comprises:

(15) storing a pointer to the shared access requestor of the local node;

(16) determining whether the shared access request of the local node is valid;

(17) terminating an exclusive access of the local node when the exclusive access request is no longer valid; and,

(18) returning information requested by the shared access request of the local node to cache.

26. (original) A method for cache coherence in a distributed shared memory (DSM) multiprocessor system comprising:

(1) receiving a first shared access request by a receiving node, wherein the receiving node contains shared information;

(2) receiving a second shared access request by the receiving node;

(3) processing one of a first and a second shared access requestor as an exclusive access request; and

(4) transmitting an invalidate instruction to other of the first and the second shared access requestor.

27. (original) The method of claim 26, wherein step (3) further comprises

(5) determining whether one of the first and the second shared access requestor originates from a local node or a remote node;

(6) processing the first shared access requestor as the exclusive access request when the first shared access request originates from the remote node;

(7) transmitting the invalidate instruction to the second shared access requestor.

28. (original) The method of claim 26, wherein step (3) further comprises:

(5) determining whether one of the first and the second shared access requestors originates from a local node or a remote node;

(6) processing the second shared access request as the exclusive access request when the first shared access request originates from the remote node;

(7) transmitting the invalidate instruction to the first shared access requestor.

29. (original) The method of claim 26, wherein the method further comprises processing as the shared access request to one of the first and the second shared access request when the other of the first and the second shared access originates from a local node.

30. (original) The method of claim 26, wherein the method further comprises invalidating an exclusive access of one of the first and the second shared access request originating from a remote node when the other of the first and the second shared access request originates from a local node.

31. (original) The method of claim 30, wherein said invalidating step further comprises:

(5) comparing upper bits of an address of one of the first and the second access requestors with upper bits of the address of the receiving node;

(6) processing one of the first and the second shared access request as the exclusive access request if the upper bits of the address of one of the first and the second shared access requestors match the upper bits of the address of the receiving node.

32. (original) The method of claim 30, wherein said invalidating step further comprises:

(5) comparing upper bits of an address of one of the first and the second access requestors with a table of addresses stored by the receiving node;

(6) processing one of the first and the second shared access request as the exclusive access request if the upper bits of the address of one of the first and the second shared access requestors match the table of addresses stored by the receiving node.

33. (original) The method of claim 32, wherein the method further comprises:

(7) storing a pointer to one of the first and the second shared access requestors;

(8) determining whether the one of the first and the second shared access request is valid;

(9) terminating an exclusive access of one of the first and the second shared access requestors when one of the first and the second shared access request is no longer valid; and,

(10) returning information requested by one of the first and the second shared access requestors to cache.

34. (original) A system for maintaining cache coherency in a distributed shared memory (DSM) multiprocessor systems, comprising:

a plurality of processing nodes, wherein each said node is attached to at least one central processing unit (CPU) and a memory unit;

at least one receiving node for receiving a shared access request;

a means for determining whether said shared access request originates from said local node or from said remote node;

a plurality of shared memory vectors, wherein each said shared memory vector defines either said local node requesting information sharing from said receiving node or said remote node requesting information sharing from said receiving node;

an exclusive memory access pointer, wherein said exclusive memory access pointer points to said remote node in the DSM multiprocessor system that has been granted an exclusive access to information in said receiving node;

a storage unit for storing said exclusive memory access pointer; and,

a deleting means for eliminating said exclusive memory access pointer, when said exclusive memory access pointer becomes invalid.

35. (original) The system of claim 34, wherein the system operates using directory-based protocol environment.

36. (original) The system of claim 34, wherein said deleting means further comprises a means for transmitting of an invalidate instruction to other said processing nodes once said exclusive memory access pointer points to said remote node in the DSM multiprocessor system.

37. (original) The system of claim 34, wherein the system further comprises a means for transmitting another invalidate instruction to said remote node when said receiving node has received said shared access request from said local node, whereby an address of said local node matches an address of said receiving node.

38. (original) A system for maintaining cache coherency using a directory-based protocol environment in a distributed shared memory (DSM) multiprocessor systems, comprising:

a plurality of processing nodes, wherein each said node is attached to at least one central processing unit (CPU) and a memory unit;

at least one receiving node for receiving a shared access request;

a means for determining whether said shared access request originates from said local node or from said remote node;

a plurality of shared memory vectors, wherein each said shared memory vector defines either said local node requesting information sharing from said receiving node or said remote node requesting information sharing from said receiving node;

an exclusive memory access pointer, wherein said exclusive memory access pointer points to said remote node in the DSM multiprocessor system that has been granted an exclusive access to information in said receiving node;

a means for transmitting an invalidate instruction to other said processing nodes once said exclusive memory access pointer points to said remote node in the DSM multiprocessor system;

a storage unit for storing said exclusive memory access pointer; and,

a deleting means for eliminating said exclusive memory access pointer, when said exclusive memory access pointer becomes invalid.

39. (original) A system for maintaining cache coherency using a directory-based protocol environment in a distributed shared memory (DSM) multiprocessor systems, comprising:

a means for receiving a shared access request for information;

a means for determining whether the shared access request originates from a local node or from a remote node;

a means for processing the shared access request as an exclusive access request when the shared access request originates from the remote node;

a means for processing the shared access request when the shared access request originates from the local node.

40. (currently amended) A system for maintaining cache coherency using a directory-based protocol environment in a distributed shared memory (DSM) multiprocessor systems, comprising:

a means for receiving a shared access request for information by a receiving node;

a means for determining whether the shared access request originates from a local node or from a remote node, wherein said local node is physically proximate to said receiving node, wherein said remote node is physically distant from said receiving node;

a means for processing said access request differently depending on whether said access request originates from said local node or from said remote node.

41. (original) A computer program product comprising a computer useable medium having computer logic stored therein, said computer program logic enabling a computer system to maintain cache coherency in a distributed shared memory (DSM) system, wherein said computer program logic comprises:

a first function that enables the computer to receive a shared access request for information;

a second function that enables the computer to determine whether the shared access request originates from a local node or from a remote node;

a third function that enables the computer to process the shared access request as an exclusive access request when the shared access request originates from the remote node; and

a fourth function that enables the computer to process the shared access request when the shared access request originates from the local node.